Team ID: Bio1

Member name	Student's ID
1. Kareem Mohamed Wardany	20191701149
2. Karim Tarek Emam	20191701148
3. Omar Mostafa Mohamed	20191701137
4. Mark Sameh William	20191701153
5. Monica Rafik William	20191701221

Layers

These are the layers that will be used to build our two architectures

Convolutional Layer:

It is the main building block of CNN. It contains a set of filters, parameters of which are to be learned throughout the training. The size of the filters is usually smaller than the actual image. Each filter convolves with the image and creates an activation map.

Max pooling Layer:

It is a pooling operation that selects the maximum element from the region of the feature map covered by the filter. Thus, the output after max-pooling layer would be a feature map containing the most prominent features of the previous feature map.

Normalization Layer:

It normalizes input across the features instead of normalizing input features across the batch dimension in batch normalization.

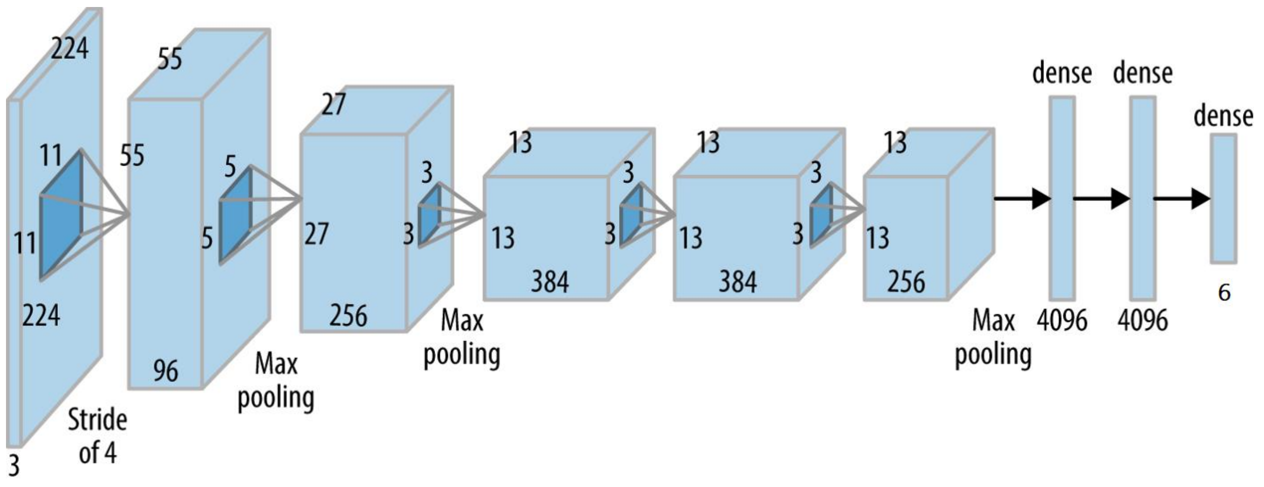
Fully connected (Dense) Layer:

It’s a layer in which each neuron applies a linear transformation to the input vector through a weight’s matrix. As a result, all possible connections layer-to-layer are present, meaning every input of the input vector influences every output of the output vector.

Dropout Layer:

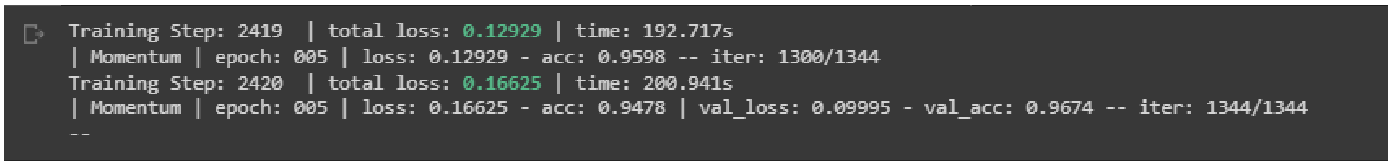
It is a mask that nullifies the contribution of some neurons towards the next layer and leaves unmodified all others.

AlexNet Architecture



- This architecture was not able to predict yoga, swimming, and rowing correctly.
- So, an improvement was made that first convolution to be (7,7) instead of (11,11) with same number of filters and stride, so architecture was able to predict yoga, swimming, and rowing better than before.

Last Training logs:

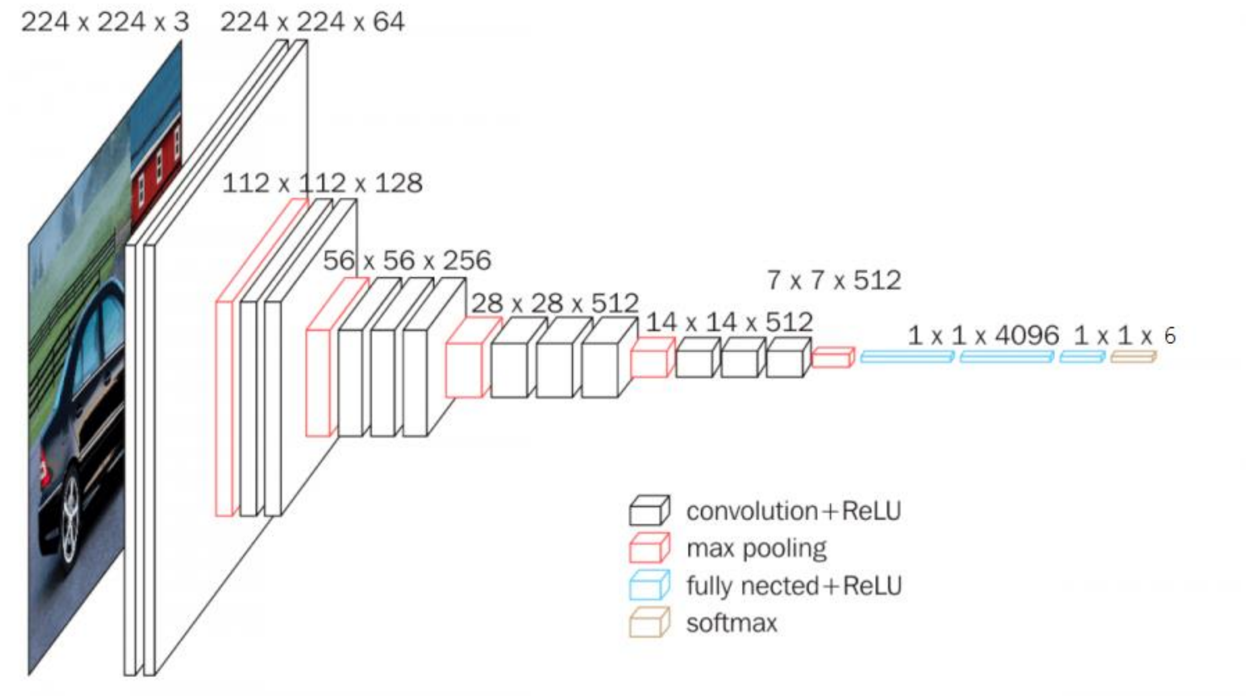


```
Training Step: 2419  | total loss: 0.12929 | time: 192.717s
| Momentum | epoch: 005 | loss: 0.12929 - acc: 0.9598 -- iter: 1300/1344
Training Step: 2420  | total loss: 0.16625 | time: 200.941s
| Momentum | epoch: 005 | loss: 0.16625 - acc: 0.9478 | val_loss: 0.09995 - val_acc: 0.9674 -- iter: 1344/1344
--
```

NoteBook:

https://colab.research.google.com/drive/1KyS2HUz7OjHO058BBwTNj85wtp-cqYeu?usp=sharing

VGG16 Architecture



- In VGG 16 architecture using Adam optimizer was poor with predictions, instead, Stochastic gradient descent (SGD) yielded better results.
- Without using dropout layer, the model didn't yield acceptable accuracy and almost overfitted the data but with using dropout layer, the model fitted the data and gave better accuracy.
- Without using Image generator, the model didn't yield acceptable accuracy and almost overfitted the data but with using Image generator, the model fitted the data and gave better accuracy.
- Comparing VGG16 to VGG19 architectures, VGG16 gave better results.

NoteBook:

https://colab.research.google.com/drive/1U-iDFTrROuKLyqYbVJAHzr35Dkjk4yVO?usp=sharing